01

Produciendo datos

Transcripción

[00:00] Bienvenidos a la cuarta clase de su curso de Spring Boot Java API. En esta ocasión vamos a ver un nuevo caso de uso, ya vimos lo que es la creación de médicos, ya sabemos validar los datos que están llegando, ahora vamos a entender cómo listarlos.

[00:18] Si ustedes recuerdan en la clase anterior para saber si el médico era guardado correctamente o no, entrábamos directamente a la base de datos y revisábamos si es que el registro había sido creado correctamente o no. En esta oportunidad ya no queremos usar más la base de datos directamente, no queremos tener interacción directa contra el motor de base de datos.

[00:38] Queremos hacerlo todo a nivel de API, para esto tenemos el requerimiento del negocio del listado de médicos, que es algo muy parecido a lo que ven aquí en la pantalla. Es solo con fines genéricos referenciales, no es tal cual, pero, por ejemplo, un típico caso de uso podría ser deseamos que en el aplicativo móvil o en la web los médicos sean mostrados y yo pueda escoger a un médico o tener la lista completa de médicos que tengo disponibles.

[01:09] Para esto tenemos una serie de consideraciones para esta lista. Por ejemplo, la información que debe ser mostrada es el nombre, especialidad el documento y el email. Entre las reglas del negocio. ¿Cuáles son? La primera debe ser ordenado ascendentemente, y esto quiere decir, yo debería ser capaz de ordenar por nombre o por código de forma ascendente.

[01:34] Nuestra vista tiene que ser capaz de organizarse de esa forma. Y paginado máximo 10, registros por página. ¿Qué significa paginado? Normalmente, cuando ustedes reciben una lista de alguna aplicación, digamos si aplicación tiene 500 registros, ustedes no reciben los 500 registros en una sola request.

[02:04] Normalmente ustedes tienen una serie de números abajo de la lista, diciendo que es la página 1, página 2, página 3, esto es para implementar mejor performance en las queries en cuanto a la base de datos, y también para una mejor experiencia de usuario. Entonces sin más que decir, vamos a comenzar con la parte interesante que es el código.

[02:27] Vimos que el RequestMapping está apuntando a médicos, significa que en el pad, en la URL apuntamos a médicos, tenemos nuestro método RegistrarMédico y ahora vamos a implementar un nuevo método llamado listar. Para esto, aquí usamos un método del tipo void, que eso quiere decir un método que no retorna parámetros, pero esta vez yo voy a usar una lista.

[02:54] ¿Por qué? Porque quiero retornar una lista de algo, pero obviamente la lista que voy a usar es la lista de Java-util, siempre, y por esta vez yo voy a retornar una lista de médico como entidad. Esto es por ahora solamente para que se entienda lo que quiero hacer, no va a quedar así finalmente.

[03:15] Pero bueno, ponemos listadoMedicos sin parámetros, no vamos a considerar ni un solo parámetro. Y si aquí usamos un postMapping para el listado, aquí vamos a usar un putMapping. Perdón, no putMapping. GetMappint, porque el request es del tipo get, vamos a un getMapping, lo lista y todo bien. ¿Qué es lo que vamos a poner ahora dentro de este método?

[03:44] Vamos a retornar una lista de médicos directamente, por lo tanto, vamos a decirle return medicoRepository.findAll() sin ningún tipo de parámetro. Y ustedes me preguntarán, pero Diego, en medicoRepository, si nosotros entramos no hay ni un solo método creado. El método findAll viene

de la clase que estamos extendiendo, de JpaRepository. Esta es la magia de las interfaces que Spring nos brinda.

[04:17] No necesitamos escribir este código porque finalmente ya Spring lo implementa por nosotros internamente, nosotros solamente tenemos que llamarlo. Y bien, ¿esto sería todo lo que necesitamos para poder retornar nuestra lista de médicos? Vamos a probarlo, entonces guardamos y ejecutamos nuestro servidor. Esperamos que compile, vemos que ya inicia.

[04:49] Inició la aplicación correctamente, y nos vamos ahora para Insomnia. En Insomnia vamos a crear una nueva request, le damos clic aquí http request y le vamos a poner de nombre listado de médicos, vamos a renombrar y seguimos con listado médicos. Perfecto. Y la url sería la misma, sería http://localhost:8080 (http://localhost:8080) y también con /medicos porque es la url que estamos siguiendo.

[05:24] Le voy a dar get porque yo estoy usando un getMapping en esta url y vamos a ver qué es lo que me he retorna. Vamos ahí y, en efecto, yo recibí aquí el listado de médicos que yo tengo en este momento. Tengo a Diego López, tengo a Diego López con otra dirección, con otro id, y a Erick Cárdenas, pero aquí entra nuestra primera constrain por regla del negocio.

[05:50] Yo estoy listando los médicos, ¿pero debería mostrar todos estos datos? Si regresamos a nuestra lista de requerimientos, lo único que debemos mostrar es el nombre de la especialidad, el documento y el email. ¿Entonces, cómo es que vamos a conseguir restringir los demás valores? Eso lo vamos a ver en el siguiente video.